A Deep Learning-Based Object Detection and Tracking System for Self-Driving Cars

Affan Idowu Kareem (Researcher), **Gabriella Lakatos (Supervisor)**

For the degree in MSc Artificial Intelligence and Robotics School of Physics, Engineering and Computer Science, University of Hertfordshire, UK

Abstract

Object detection for autonomous cars is a challenging task with many applications. Accurate detection of objects in real-world scenarios, such as car and traffic detection, is crucial to ensuring road safety, reducing accidents, and improving the efficiency of transportation systems. One of the most promising approaches to this problem is using deep learning-based object detection algorithms. You Only Look Once (YOLO) is a cutting-edge one-stage detector with remarkable results in detecting objects in real-time images. This study employed YOLOv3 and YOLO-R to detect street-level objects in real-time driving scenarios. The algorithms were trained using a modified Udacity Self-Driving Car Dataset comprising 15,000 images. The dataset was meticulously pre-processed and augmented. After selecting the algorithms, the next step was to train and evaluate each on the dataset. The experiments yielded impressive results for both algorithms, each achieving high Precision, recall, F1-Score, and Mean Average Precision (mAP) values that exceeded expectations. Upon comparing the performance of the algorithms, it was found that YOLOv3 demonstrated superior accuracy with a mAP of 0.66, outperforming YOLO-R, indicating that it can effectively detect street-level objects with high Precision and recall. Furthermore, the study reveals that YOLOv3 is highly efficient in detecting occluded objects in different weather conditions, making it a viable candidate for real-world applications.

Keywords: Object Detection, One-stage Detectors, YOLO, Self-Driving Cars

1.0 Introduction

Self-driving cars are becoming increasingly prevalent today, aiming to improve road safety and efficiency. However, one major challenge in achieving this goal is accurately detecting and tracking objects in the environment. This study addresses this challenge by developing a deep learning-based object detection and tracking system for driverless cars. Recent advancements in deep learning have significantly improved object detection and tracking performance. According to Ren et al. (2015) and Liu et al. (2016), Convolutional Neural Networks (CNNs) have been employed for detecting objects, and Long Short-Term Memory (LSTM) have been used for tracking objects (Wojke et al., 2017; Bewley et al., 2016). However, these methods have limitations when applied to self-driving cars.

As self-driving car technology advances, deep neural network techniques have become critical in ensuring safe and efficient autonomous driving. Thanks to the large repository of datasets and the rapid development of deep learning techniques, it is now possible to develop highly accurate and reliable object detection and tracking algorithms. These algorithms can assist self-driving cars in navigating complex and dynamic environments where traditional object detection and tracking approaches may not suffice. These algorithms rely on sensors such as cameras, lidar, and radar to capture data about the environment, which is then processed using computer vision algorithms to detect and track objects in real-time.

One of the most popular deep learning-based object detection and tracking systems used in self-driving cars is the YOLO (You Only Look Once) algorithm. YOLO is an object detection algorithm that divides an image into a grid of cells and predicts the bounding box and class probability for each cell (Redmon et al., 2016). This innovative approach allows YOLO to achieve high accuracy while maintaining real-time performance, making it well-suited for self-driving car applications. By quickly and accurately detecting objects in an image, YOLO can help self-driving cars make informed decisions in real-time, enhancing their ability to navigate safely and efficiently.

Another popular deep learning-based object detection algorithm in self-driving cars is Faster R-CNN (Ren et al., 2015). Faster R-CNN has been shown to achieve forefront performance on object detection benchmarks and has been used in several self-driving car systems (Chen et al., 2018). Object tracking algorithms use the information obtained from object detection algorithms to track objects over time, allowing the vehicle to predict their future behaviour and take appropriate actions. The Siamese network is a popular deep-learning object-tracking algorithm used in self-driving cars (Bertinetto et al., 2016). Siamese networks are neural networks that use a shared feature extractor to compare the appearance of two objects and predict their similarity score. This approach demonstrates high accuracy and robustness in object-tracking applications.

1.1 Video Tracking and its challenges

Video tracking, which plays a vital role in self-driving cars, is a complex and rapidly evolving field constantly pushing the boundaries of computer vision technology (Zhang et al., 2021). One of the biggest challenges in video tracking is occlusion, which occurs when objects are partially or entirely hidden from view by other things in the scene (Wei et al., 2021); this can cause tracking algorithms to lose track of objects or even mistakenly associate different objects with each other. To address this challenge, researchers have developed sophisticated algorithms that use context, motion models, and other cues to infer the presence and location of occluded objects (Zhang et al., 2021).

Another critical area of research in video tracking is the development of methods that can simultaneously handle multiple types of objects, including people, vehicles, animals, and other things (Zhang et al., 2021); this requires accurate detection and tracking algorithms and methods for classifying and understanding the behaviour of different types of objects.

Recent progress in video tracking owes much to the advancements in machine learning and deep learning techniques (Deng et al., 2021). These techniques have enabled researchers to develop more accurate and efficient object detection algorithms and more sophisticated tracking algorithms that can handle complex scenes with multiple objects and occlusions.

As video tracking continues to evolve, we will likely see even more sophisticated algorithms and applications emerge in the coming years (Deng et al., 2021). With the increasing availability of large datasets and powerful computing resources, it is an exciting time to work in this field.

However, developing accurate and reliable object detection and tracking systems for self-driving cars is still challenging. One of the significant challenges is dealing with complex and dynamic environments that contain a wide range of objects and scenarios. Another challenge is ensuring the robustness and reliability of the algorithms in the face of sensor noise, occlusions, and other sources of uncertainty.

In addition, the need for real-time performance is another major challenge in using deep learning techniques for driverless cars. Driverless cars require real-time data processing from sensors and algorithms to make quick decisions and respond to environmental changes. Therefore, developing algorithms that can process data quickly and efficiently is crucial. This has led to the development of various techniques, such as network pruning and quantisation, to minimise the computational intricacy of deep-learning models without compromising accuracy (Han et al., 2015).

Furthermore, when developing deep learning-based object detection and tracking systems for self-driving cars, it is crucial to consider the legal and ethical implications. While self-driving cars can potentially reduce accidents caused by human error, they also raise significant ethical concerns regarding liability and responsibility in the event of accidents. Researchers and policymakers must consider these issues and develop frameworks for regulating and ensuring safe and ethical self-driving car technology.

Researchers are exploring various techniques and approaches to address these challenges, including multi-sensor fusion, domain adaptation, and reinforcement learning (Wang et al., 2021).

In summary, deep learning-based object detection and tracking systems are essential for self-driving cars to operate safely and effectively in complex and dynamic environments. With the increasing availability of large-scale datasets, there is great potential for further advancements in this field. However, developing robust and reliable algorithms that can handle the complexities of the natural world remains a significant challenge and will require continued research.

1.2 Aim of Study

This study aims to create a deep learning-powered object detection and tracking solution that can effectively and accurately identify and track vehicles, pedestrians, and obstacles in real-world driving situations, aiming to enable their use in self-driving cars. The study involved testing and evaluating various models using the Udacity dataset, focusing on performance metrics such as Precision, F1 score, recall, and mean average Precision. In this study, we propose using YOLO as the basis for our object recognition models because it can rapidly detect objects, which is critical for object-tracking tasks requiring a high recognition rate.

1.3 Research Gap and Justification

The problem addressed by Deep Learning-Based Object Detection and Tracking Systems for Self-Driving Cars is to accurately detect and track objects in the surrounding environment of a self-driving car. The goal is to provide the self-driving car's decision-making system with reliable and real-time information about the location and movement of objects such as vehicles, and pedestrians, including small children, animals, and cyclists. This information is crucial for the self-driving car to make safe and effective driving decisions. Traditional object detection and tracking methods have limitations in accuracy and efficiency, particularly in complex and dynamic driving scenarios. Deep learning-based approaches, such as convolutional neural networks (CNNs) and tracking-by-detection methods, have shown promising results in improving object detection and tracking accuracy and efficiency for self-driving cars. However, challenges still need to be addressed, such as handling occlusions, detecting small or distant objects, and ensuring real-time processing.

1.3.1 Research Questions:

From the research gap, this study tends to work on a system that checks the challenges identified and provides accurate and reliable object detection and tracking capabilities for self-driving cars in real-world driving scenarios. Hence, the following research questions are to be answered:

1. Can deep learning techniques improve the performance of self-driving car?
2. Can the proposed system handle small and occluded objects, changes in lighting and weather conditions, and can it be robust to these variations?
3. How does the proposed system compare to state-of-the-art object detection and tracking methods regarding performance and real-time capabilities?

1.3.2 Specific Objectives of the Study:

To answer the research questions, the following objectives are set for the study:

1. To develop and optimise deep learning models for object detection and tracking to handle challenging driving scenarios, such as low light, heavy traffic, and adverse weather conditions.

2. To evaluate the performance of the developed system through extensive testing and validation using performance and evaluation metrics.

3. To compare the developed system against existing state-of-the-art object detection and tracking systems for self-driving cars and identify areas for further improvement.

The next section provides a thorough analysis of the pertinent ideas and supporting literature as well as more general background data on how object detection and tracking operate. The third section outlines the technique utilised in this study as well as the criteria used to evaluate the findings. The experiment's findings are provided in the fourth part. The most effective model(s), according to the utilised evaluation metrics, are presented in the fifth part, which also examines the findings from the preceding section. Finally, the sixth section brings the research to a close and makes suggestions for additional research. It summarises and evaluates the report, including whether the primary aims of the research have been achieved, and identifies areas for further investigation.

2. Related Works

Studies into various object recognition and tracking techniques have increased recently as a result of the growing use of machine learning technologies. Object tracking and object detection are the two main areas that research are centred on. Researchers in these fields have put forth and evaluated a variety of strategies. Theoretical ideas, related research, and important topics that are important to this dissertation will all be covered in this section.

2.1 Object Detection

Object detection involves recognising and categorising objects in an image. Modern object detection algorithms widely use convolutional neural networks (CNNs). These algorithms can be classified into two main categories: single-stage and two-stage object detectors. Two-stage object detectors identify regions of interest in the image and organise them separately, while single-stage detectors simultaneously predict the bounding boxes and object classes. Although single-stage detectors have faster processing speed, they may sacrifice accuracy compared to region-based object detectors.

There has been notable progress in developing more accurate single-stage detectors, such as the YOLOv4 detector (Bochkovskiy et al., 2020), achieving forefront performance while maintaining real-time processing speed.

Furthermore, recent advancements in object detection techniques have also led to the development of novel approaches, such as anchor-free detection, which eliminates the need for predefined anchor boxes and can potentially achieve a higher detection accuracy (Zhou et al., 2019). The balance that exists involving swiftness and precision requirements determines which object detection method is best for a particular application. With continued research and development in this field, future advancements are likely to result in more accurate and efficient object detection algorithms.

Object classification is a crucial task for autonomous vehicles, as it involves identifying the class instances to which an object belongs, determining their precise location, and taking appropriate actions accordingly. This falls under the subdomain of computer vision and is essential for achieving the goals of self-driving cars, including increasing safety, minimising accidents, and making transportation more efficient and reliable.

Numerous methods exist to process gathered data and provide actionable insights for autonomous vehicles. For instance, the GTI and KITTI datasets utilise the HOG feature extraction algorithm along with various classification techniques to identify multiple cars in images. An alternative approach involves using the "You Only Look Once" (YOLO) algorithm for real-time object detection, which identifies specific objects in videos. Castello et al. (2020) evaluated the YOLO version 3 and 4 by training them on a recent, large-scale Berkeley Deep Drive (BDD100K) dataset. They reported that YOLOv4 using MISH and SWISH functions, performed better than the initial version of the Leaky ReLU. Mobahi and Sadati (2020) also utilised the BDD100K dataset for training and testing algorithms using Python and PyTorch, classifying objects into three scales (small, medium, and large) utilising the Faster R-CNN,

single-shot multi-box detector (SSD), and PyTorch algorithms. They found that their PyTorch implementation improved object classification accuracy, particularly for small-scale objects.

Mirza et. al., (2021) compare the performance of three object detection and classification methods on road networks: YOLOv3, PointPillars, and AVODS. Results showed that PointPillars achieved the highest mean average Precision (mAP) on night scenes in the KITTI dataset but failed in adverse weather conditions such as rain. Mai et al. (2021) trained the Spare LiDAR Stereo Fusion Network (SLS-Fusion) on the KITTI dataset to address this issue, improving object classification in foggy weather. In addition, Al-refai and Al-refai (2020) utilised the YOLOv3 algorithm with Darknet-53 CNN to detect and classify objects such as cars, trucks on the road network. They found that YOLOv3-Darknet had a high wrong classification rate for small things. Furthermore, Grigorescu et al. (2020) proposed a deep convolutional neural network to extract image representations automatically and use it in the perception system of autonomous vehicles to identify and recognise objects on the road.

According to Masmoudi et al. (2019), integrating LiDAR point cloud and CNN-equipped cameras can enable the recognition of pedestrian and vehicle positions. Additionally, Lidestam et al. (2020) demonstrated that deep learning can process sounds from emergency vehicles and determine their direction from a distance. Meanwhile, Agafonov and Yumaganov (2020) used both genuine and artificial data to employ object categorization and 3D object detection, explicitly using the KITTI dataset to evaluate the detection of vehicles. They utilised the open-source simulator CARLA to generate synthetic data and train autonomous vehicle control systems for testing various traffic scenarios. The average Precision of classification objects was employed as an evaluation metric. They did find a constraint in the ability to detect things using real-world data for 3D recognition and categorization of objects techniques that had only been trained on synthetic data.

2.2 Multiple Object Tracking

In tracking-by-detection methods, object tracking involves predicting the future positions of objects and associating newly detected objects with already tracked objects based on their predicted positions. The Kalman filter is commonly used for predicting the future positions of objects, and the Hungarian method is used to solve the assignment problem of associating objects in new frames with already tracked objects. One example of such a system is Deep SORT, which uses the Kalman filter and the Hungarian method for prediction and assignment. It also incorporates a 128-dimensional visual descriptor obtained by feeding the object's bounding box into a CNN (Wojke et al., 2017).

The use of CNN-based visual descriptors has significantly improved object tracking accuracy in recent years. For instance, in the Deep SORT system, the CNN is trained to distinguish between pedestrians based on their visual features, allowing for better tracking of people in crowded scenes (Wojke et al., 2017). In addition, CNN-based object detection algorithms such as YOLO and Faster R-CNN have demonstrated outstanding performance on challenging object detection tasks. These algorithms employ a CNN to extract features from an image and predict the location of objects in the image (Redmon et al., 2016; Ren et al., 2017).

Recently, there has been an increased interest in integrating object detection and tracking for self-driving cars. The goal is to use object detection and tracking to provide accurate information about the surrounding environment to the self-driving car's decision-making system. For example, the YOLOv4 object detection algorithm has been employed in

developing a self-driving car to detect and track objects such as pedestrians and vehicles (Bochkovskiy et al., 2020). The Deep SORT algorithm has also been utilised in a self-driving car to track objects such as other vehicles, pedestrians, and cyclists (Tao et al., 2021).

Object detection and tracking are crucial for various applications, including self-driving cars. Deep learning-based approaches, such as those utilising CNN-based visual descriptors, have shown remarkable performance improvements in recent years. With further advancements in deep learning techniques and computing capabilities, object detection and tracking systems are expected to become even more accurate and efficient. The proposed approach in this study aims to present significant contributions by utilising recent advancements in neural networks for tracking and detecting objects in real-time.

2.3 Deep Learning for Computer Vision

Cameras serve as a primary passive sensor for environment perception in autonomous vehicles, while lidar, sonar, and radar serve as active sensors. Although other sensors are available, cameras are more affordable and accessible, making them the focus of research on environment perception. Deep learning and computer vision are essential in achieving precise and accurate perception processes and mapping and localisation using cameras. The two most used types of cameras are monocular cameras which can extract detailed information regarding shapes and textures using pixel intensities, they cannot accurately estimate an object's position and size due to the lack of depth information, and the other is the stereo camera (Arnold et al., 2019). On the other hand, stereo cameras can solve this issue by providing additional depth information.

Atmospheric conditions can significantly impact camera beams reliability and accuracy. Snowy or rainy weather can make calculations from images more complex, while night-time can cause issues in calculating depth using camera images. Some research proposes methods to detect vehicles from in-vehicle monocular cameras during night-time (Kosaka and Ohashi, 2015). However, lidar remains the most reliable and accurate sensor, even during night-time. Lidar provides precise distance measurements and is unaffected by light conditions.

Nonetheless, the cost of lidar is one of the reasons why autonomous vehicles are expensive. Therefore, while cameras are not 100% reliable, they are still widely used in autonomous vehicles due to their availability and lower cost. In contrast, lidar is combined with cameras to provide accurate and reliable perception.

Detecting pedestrians on the road is a critical task for autonomous vehicles as it directly relates to the safety of individuals. To address this challenge, Chen and Huang, (2019) proposed a combination of an RGB-D stereo-vision and a thermal camera for pedestrian detection. The study evaluated the HOG and Convolutional Channel Features (CCF) based detection methods on a multispectral dataset, with CCF outperforming HOG regarding pedestrian detection accuracy. In a separate study, Amanatiadis et al. (2018) developed a monocular vision-based method for detecting passengers in nearby vehicles, utilising a series of techniques, including convolutional neural networks.

To ensure safe navigation, detecting the drivable path is crucial for autonomous vehicles to see objects on the road. Amanatiadis et al. (2018) introduced a detection model named RPP (residual network with pyramid pooling), combining techniques such as convolutional

network, residual learning, and pyramid pooling for monocular vision-based road detection. However, intersections pose a significant challenge for autonomous vehicles, as detecting other vehicles or road agents is critical for safe navigation. Yudina et al. (2019) proposed a training technique that integrates reinforcement learning and computer vision, allowing unmanned vehicles to learn the behaviour of other road agents in intersections using visual information extracted from aerial photographs. This approach enhances autonomous vehicles' perception and safe positioning in road intersections.

2.4 Image Classification Networks

Image classification networks, convolutional neural networks (CNNs), are deep learning models designed to perform image recognition tasks. Image classification aims to identify objects or patterns in images and classify them into specific categories. This task has numerous applications, ranging from facial recognition and object detection to medical image analysis and self-driving cars. CNNs are a neural networks specifically designed for image processing Simonyan and Zisserman, (2014). The commonly used image classification model include:

- **VGG-16**

There are 16 completely connected convolutional layers in the VGG-16 network. The first 13 layers are convolutional, and the number of filters in each layer increases from 64 to 512, with the parameters reaching 138 million. The last three layers are fully connected, with 4096 units each, followed by a softmax activation function that produces the final classification output Simonyan and Zisserman, 2014). The VGG-16 architecture is known for its simplicity and elegance. It has a fixed input size of 224x224 pixels, and the same convolutional filter size is used throughout the network. This simplicity has made it easy to implement and train, and it has become a benchmark for comparison with other deep learning models.

- **ResNet-50**

ResNet-50 consists of 50 layers, including convolutional layers, max-pooling layers, fully connected layers, and residual blocks. The architecture is based on skip connections, which allow information to be passed from one layer to another without being processed by intermediate layers (He et al., 2015). This helps to mitigate the vanishing gradient problem, which can occur in deep networks when the gradients become too small to update the weights effectively.

- **Darknet-53**

The architecture of Darknet-53 consists of 53 convolutional layers, which is why it is named as such. The initial layers of the network perform basic feature extraction, while the subsequent layers detect more complex features. The network uses residual connections to overcome the vanishing gradient problem and facilitate training of deeper networks (Redmon and Farhadi, 2018).

One notable aspect of Darknet-53 is the use of 1x1 convolutional layers, which helps reduce computational complexity and parameter count, leading to faster training and inference times. The network also uses batch normalisation and LeakyReLU activation function, which help to prevent overfitting. Compared to previous versions of Darknet, such as Darknet-19 and Darknet-21, Darknet-53 has shown to be more accurate and efficient in object detection tasks. It achieves forefront results on the ImageNet classification task, with a top-1 error rate of 20.2%

and a top-5 error rate of 5.8% (Redmon and Farhadi, 2018). Darknet-53 has been widely used as a backbone architecture for object detection systems, including YOLOv3 and YOLOv4. It has also been used for other computer vision tasks, such as image classification and semantic segmentation. Additionally, Darknet-53 has been optimised for embedded systems such as NVIDIA's Jetson platform, making it useful for real-time applications.

2.6 Object Detection Algorithm

This section presents a theoretical introduction to the object detection algorithms used in this Dissertation and literature on other object detection algorithm. The preceding algorithms of the tested ones are also explained to facilitate a better understanding of the tested algorithms.

- **RCNN**

The RCNN framework operates in two stages: region proposal and object classification. The first stage involves generating a set of region proposals using selective search, which generates a hierarchy of regions based on their texture, color, and intensity similarities. This hierarchy is then used to identify regions of interest, which are used for further processing in the second stage (Girshick et.al., 2014), while the regions of interest are passed through a convolutional neural network (CNN) that extracts a set of features for each region in the second stage. These features are then fed into a set of support vector machines (SVMs) that classify the region into one of the object classes or background.

However, RCNN suffered from high computational costs, as the framework needed to perform a separate forward pass for each region proposal, making it slow and impractical for real-time applications (Girshick et.al., 2014). This led to faster variants such as Fast R-CNN, Faster R-CNN, and Mask R-CNN.

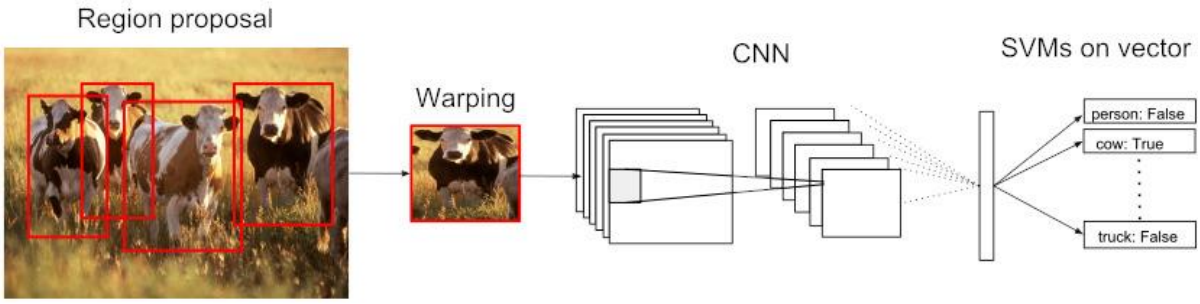

Figure 2.1: Schematic of the R-CNN pipeline.

- **FAST RCNN**

Fast R-CNN is an object recognition algorithm that improves the efficiency and precision of R-CNN. Fast R-CNN successfully recognises objects in images by merging region proposal and classification into a single model. It generates prospective object areas using region proposal methods, performs RoI pooling for spatial alignment, and applies deep learning strategies for classification and bounding box regression (Girshick et al., 2015).

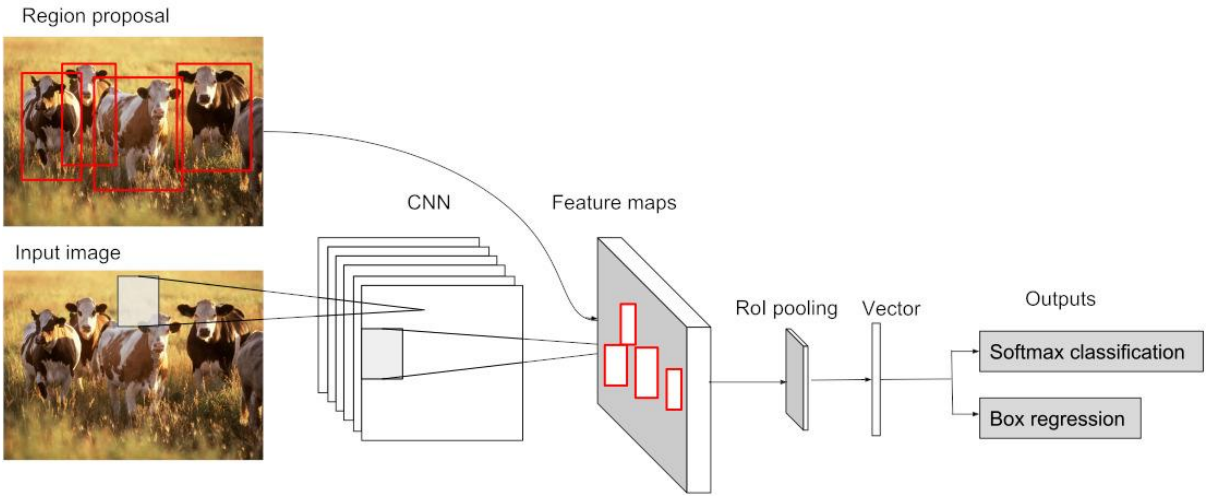

Figure 2.2: Schematic of the Fast R-CNN pipeline.

- **FASTER RCNN**

Faster R-CNN incorporates RPN into the model, building on the success of Fast R-CNN. Faster R-CNN produces even faster and more precise results by sharing convolutional characteristics across region proposal and object detection. It does away with the requirement for external region proposal techniques, making the entire detection process trainable from beginning to the end. Faster R-CNN has been a popular option for high-performance object identification jobs because of its increased efficiency and integration (Ren et. al., 2015).

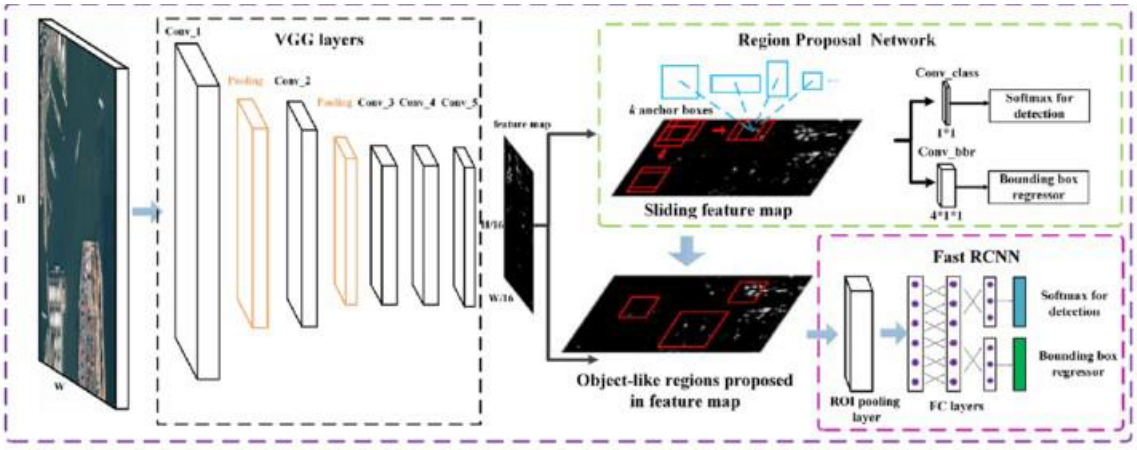

Figure 2.3: Faster R-CNN model architecture (Zhou et al., 2018)

- **YOLO**

A popular real-time object detection method for image and video analysis is called YOLO (You Only Look Once). Redmond et al. (2016) first introduced YOLO, and it has since undergone several changes.YOLO uses a deep convolutional neural network (CNN) architecture, usually with several convolutional layers followed by fully connected layers. YOLO is fast and efficient, requiring only a single forward pass through the network to make predictions. Using backpropagation, the architecture is trained end-to-end on a large, labelled dataset, such as COCO or Pascal VOC (Redmon et al., 2016). One major advantage of YOLO is its speed, as it can process up to 45 frames per second on a single GPU, making it ideal for real-time

applications. YOLO is also known for its high accuracy, as it can detect a wide range of objects with high Precision and recall. Another advantage of YOLO is its ability to detect objects at different scales and orientations, making it effective for complex scenes.

There have been several versions of YOLO, with YOLOv5 being the latest and most popular version. YOLOv5 uses a more efficient backbone architecture, a custom anchor box configuration, and a range of other improvements to achieve even better performance than its predecessors (Redmon et al., 2016).

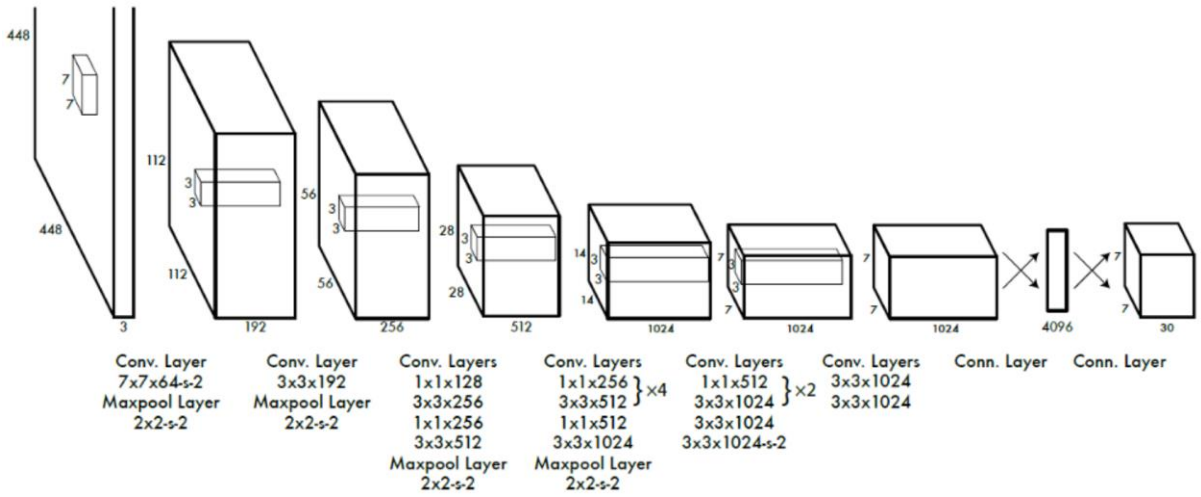

Figure 2.4: YOLO's model architecture (Sik, 2018)

- **YOLOv3**

Deep convolutional neural network (CNN) architecture is used by YOLOv3 to identify objects in images. The network consists of 53 convolutional layers and was developed using the ImageNet dataset, which includes more than a million photos and hundreds of different item classifications (Redmon and Farhadi, 2018). Skip connections and residual connections are also used in the YOLOv3 design to enhance network performance.

The YOLOv3 algorithm can detect over 80 different object categories, including people, animals, vehicles, and household items, with high accuracy and speed. YOLOv3 achieves this through a combination of two main techniques: anchor boxes and feature pyramid networks (Redmon and Farhadi, 2018). Anchor boxes are pre-defined bounding boxes of different sizes and aspect ratios that the algorithm uses to predict the position and size of objects in an image. By using anchor boxes, YOLOv3 can accurately detect objects of varying sizes and shapes. Feature pyramid networks (FPN) are used to extract features from different levels of the network and combine them to create a multi-scale feature map. This allows YOLOv3 to detect objects at different scales and resolutions, improving its overall accuracy.

Another key feature of YOLOv3 is its use of a technique called "multi-scale training". This involves training the network on images of different sizes and resolutions, which allows it to detect objects at different scales and sizes (Redmon and Farhadi, 2018).

2.5 Tracking Algorithm

- **Kalman filter**

The Kalman filter is a mathematical algorithm used for tracking and estimating the state of a system. The filter is based on a set of equations that can be used to predict the state of a system over time, given a set of observations. It was first proposed by Rudolf Kalman in 1960.

The Kalman filter works by using a set of equations that describe the system being tracked, along with a set of observations of that system. The filter then uses these equations and observations to estimate the state of the system at any given time (welch, 2020). The state of the system is represented as a set of variables that describe the system's position, velocity, and other relevant properties.

The Kalman filter is a recursive algorithm that operates in two phases: prediction and update. In the prediction phase, the filter uses the equations of motion to predict the state of the system at the next time step. In the update phase, the filter uses the observations to correct the predicted state and update the estimate of the system's state (Grewal and Andrews, 2010).

- **SORT**

SORT (Simple Online and Real-time Tracking) is designed to track objects in real-time in a cluttered and crowded environment. The algorithm is based on two stages: detection and tracking. The first stage uses object detection algorithms to detect objects in each frame of a video stream (Wang et al., 2019). The detected objects are then assigned a unique identifier based on their location, dimension, and appearance. In the second stage, SORT uses the Kalman filter to track each object over time by predicting its position and velocity based on previous measurements. The predicted state is then updated with the current measurements to refine the position and velocity estimates (Bewley et al., 2016).

- **MOSSE**

MOSSE (Minimum Output Sum of Squared Error) is a correlation filter-based tracking algorithm that learns an object template and then correlates it with the subsequent frames to locate the object. It uses a Fast Fourier transform (FFT) to compute the correlation, making it computationally efficient. However, MOSSE may struggle when the object undergoes significant scale changes or rotation (Bourennane, 2022)

- **KCF**

KCF (Kernelized Correlation Filter) is another correlation filter-based tracking algorithm that improves upon MOSSE by introducing kernelisation. KCF learns a more complex model of the object, which enables it to handle changes in scale and rotation more effectively. It also uses the FFT to compute the correlation, making it fast and efficient (Bourennane, 2022).

- **TLD**

TLD (Tracking-Learning-Detection) is a tracking algorithm that combines tracking, learning, and detection to track objects robustly. TLD uses a detector to locate the object in the initial frame, then tracks it using a combination of correlation filter-based tracking and optical flow. TLD also continually updates its model of the object, which enables it to adapt to changes in appearance and handle occlusion (Bourennane, 2022).

3. Methodology

This chapter explains the methodology used for tests and evaluations in the study. It begins with a data collection section, describing how test data was collected and the data preparation section.

3.1. Data Collection

The Udacity driving dataset used in this study was obtained from Udacity's self-driving car simulator. The dataset consists of images captured from centre, left, and right angles. Each image is accompanied by a steering angle, which indicates the direction the car should be turning. In addition to the steering angle, the dataset includes other metadata such as speed, throttle, and brake values.

The dataset contains a total of 97,942 labels, covering 11 different classes. The classes include objects such as pedestrians, bicyclists, cars, and traffic lights, among others. The dataset consists of 15,000 images, providing a rich data source for training a deep learning model.

To prepare the dataset for use in the project, it was exported using the Roboflow platform. Roboflow is a popular tool for managing and pre-processing computer vision datasets. It provides a wide range of features to ensure the dataset is properly prepared for training a deep learning model, including image resizing, data augmentation, and data format conversion.

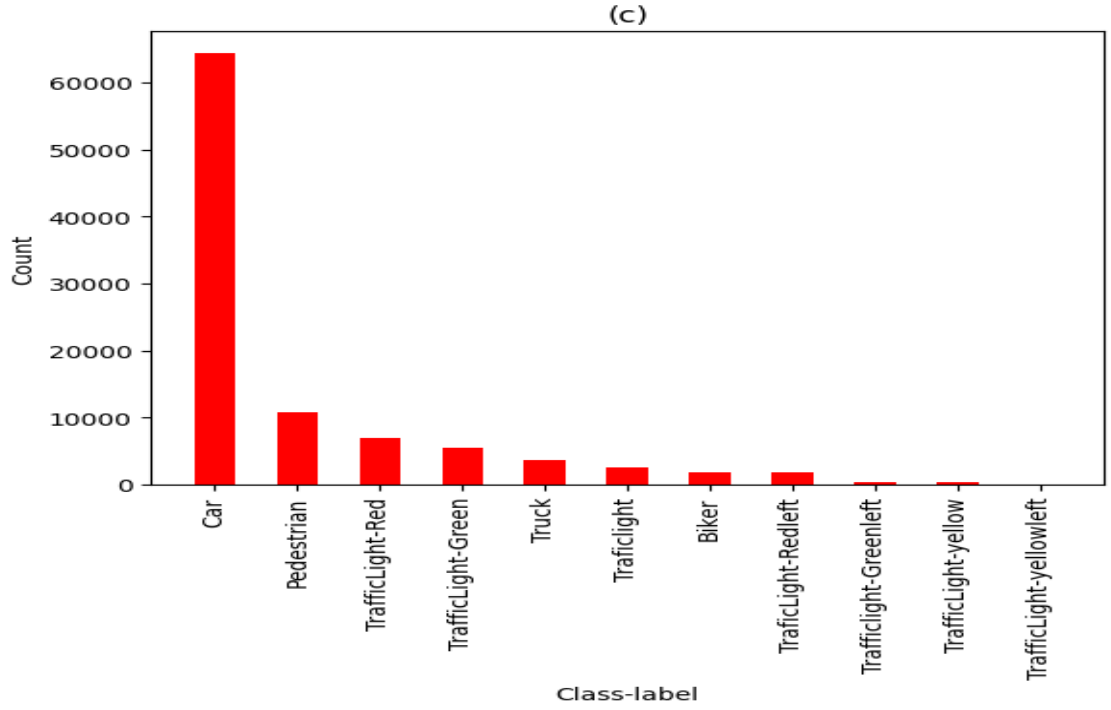

Figure 3.1: Data distribution across the class label

3.2 Data Preparation

To achieve more precise results in the analysis of the Udacity dataset, a series of data pre-processing and cleaning techniques were implemented. The "Traffic Light" class combined various traffic light categories to improve dataset accuracy and reliability. Additionally, pre-processing methods were applied to handle missing data and remove duplicated values, ensuring a balanced and high-quality dataset. Further, the dataset was examined to eliminate classes that did not have significant representation in real-world scenarios, such as other

persons, vehicles, trains, and trailers. This approach helped refine the data and increase its usefulness in the analysis context.

More accurate and reliable results were achieved with these cleaning and pre-processing techniques. The dataset's real-world images offer an exceptional opportunity to explore complex scenarios within constrained environments, providing valuable insight into the challenging tasks involved in autonomous cars. Overall, the efforts in pre-processing and cleaning the Udacity dataset have helped to ensure that the results are reliable and informative, contributing to the success of this project.

3.3 Model Training and Environment

The YOLOv3 and YOLOR models used in this study were trained on Google Colab, a cloud-based notebook environment, using a premium GPU for faster and longer runtime. The runtime environment was configured to utilise the GPU, enabling faster training of the models.

To configure the YOLOv3 and YOLOR models, their respective configuration files were modified. These configuration files contain the model architecture, hyperparameters, and other settings. Additionally, the configuration files were edited to specify classes in the dataset and their corresponding labels. This customisation enabled the models to be trained specifically for the task.

3.4 Model Evaluation

Object detection models are critical components in various computer vision applications. Evaluating the performance of these models is crucial to ensure their effectiveness in detecting objects accurately. In general, the performance of an object detection model is measured by comparing the predicted bounding boxes with the ground truth bounding boxes.

3.4.1 Classifying Boundary Boxes

Accurately classifying bounding boxes is crucial for evaluating the effectiveness of deep learning techniques in self-driving cars. Bounding boxes with an intersection over union (IoU) as calculated in equation (3.1) greater than 0.5 with a ground truth box are considered true positives (TP), while false positives (FP) lack a corresponding ground truth box, and false negatives (FN) are ground truth boxes not detected by the algorithm. In this context, true negatives (TN) are irrelevant. Notations for TP, FP, FN, and GT are used as shorthand in subsequent performance metric definitions.

$$IoU = \frac{|P \cap G|}{|P \cup G|} = \frac{|P \cap G|}{|P| + |G| - |P \cap G|} \tag{3.1}$$

3.4.2 Object Detection Evaluation metrics

The most fundamental performance metric for object detection is the classification of bounding boxes. Below are metrics used for performance evaluation in this study:

- **Recall**

In object detection evaluation metrics, recall is an important measure that indicates the ability of an algorithm to detect all ground truth objects. Recall is a measure of how well the algorithm can retrieve objects of interest. A higher recall value implies that the algorithm has detected more objects of interest, while a lower recall value suggests that the algorithm has missed some objects (Lin et al., 2017).

$$Recall = \frac{TP}{TP+FN}.100 \quad (3.2)$$

- **Precision**

Precision as an object detection evaluation metric, quantifies the accuracy of the predicted bounding boxes. The precision value reflects the ability of the algorithm to accurately identify the objects of interest, among all the bounding boxes it has predicted (Lin et al., 2017).

$$Precision = \frac{TP}{TP+FP}.100 \quad (3.3)$$

- **F1 Score**

The F1 score is a performance metric that amalgamates both recall and Precision, thereby providing a single score that quantifies the effectiveness of an object detection algorithm. This score is calculated by taking the harmonic mean of Precision and recall (Lin et al., 2017).

$$F1 = \frac{2TP}{2TP+FP+FN}.100 \quad (3.4)$$

- **Average Precision**

The Average Precision (AP) evaluation metric is determined by calculating the area beneath the precision-recall curve, which is generated by arranging all predictions in descending order based on their confidence score. Precision and recall are then calculated iteratively at various ranks in the ordered set of predictions. Figure 3.2 illustrates an instance of the precision-recall curve, with five predicted objects and three ground truth objects. The AP score is obtained by computing the area beneath this curve (Lin et al., 2017).

We can calculate AP for each class using equation (3.5)

$$Ap_n = \frac{1}{GTP} * \sum_{k=1}^{n} p(k)\Delta r(k) \quad (3.5)$$

Where;
p(k) = Precision at k ;
r(k) = relevance function at k.
The mAP is the summated average of all the APs calculated and computed as equation (3.6)

$$mAP = \frac{\sum_{n=1}^{N} AP_n}{N} \quad (3.6)$$

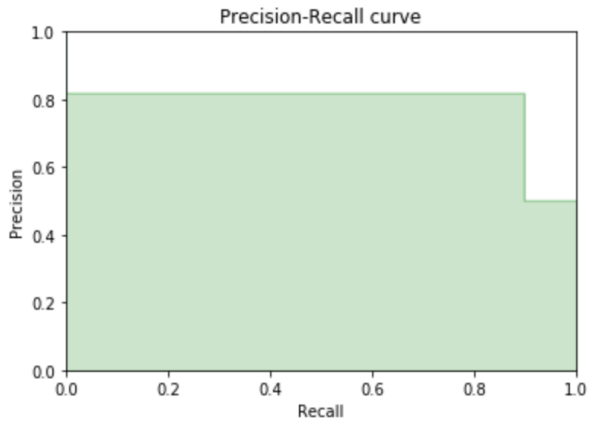

Figure 3.2: Precision-recall curve for example predictions.

- **Multiple Object Tracking Precision (MOTP)**

MOTP measures the average overlap between true positive bounding boxes and their corresponding ground truth boxes. It is calculated using equation (3.5)

$$tracking\ Precision = \frac{the\ number\ of\ correct\ tracking\ ID}{the\ number\ of\ ground\ truth} \quad (3.5)$$

3.5 Deep learning libraries

Deep learning libraries are software frameworks designed to make simpler the process of building, training, and deploying deep neural networks. These libraries provide a wide range of pre-implemented algorithms, data structures, and optimisation techniques that help developers create high-performance machine learning models.

- **TensorFlow**

TensorFlow is a popular open-source library for machine learning and artificial intelligence applications, which allows users to build and train various types of machines learning models, including deep learning neural networks, on a range of devices, from CPUs to GPUs and even TPUs (Tensor Processing Units). A major feature of TensorFlow is its computational graph, which allows users to define complex operations as a series of interconnected nodes. These nodes represent mathematical operations that can be combined to form a complete neural network model (Chanho et. al., 2018).

- **PyTorch**

PyTorch is an open-source machine learning library based on the Torch library, which is implemented in Lua, but PyTorch uses Python as its primary interface. PyTorch offers two main features: tensor computation (like NumPy) with strong GPU acceleration and building deep neural networks on a tape-based autograd system (Paszke et. al., 2017). This allows for rapid prototyping and experimentation, as well as efficient computation on both CPUs and GPUs.

- **Keras**

Keras is a high-level neural network API that is written in Python and can run on top of TensorFlow, CNTK, or Theano. It was developed to enable fast experimentation with deep neural networks and offers a user-friendly interface for creating, training, and evaluating various types of deep learning models. Keras provides a range of pre-built neural network layers, such as convolutional layers, recurrent layers, and dense layers, which can be easily assembled to create complex neural network architectures. It also supports various activation functions, loss functions, and optimisers, making it easy to configure and fine-tune models (Chollet et al. 2015).

3.6 Legal, Ethical and Social Issues:

The legal, ethical, and social implications surrounding autonomous cars are major concerns that must be considered. Several of these implications are highlighted below:

3.6.1 Legal Issues

- Liability concerns: In the event of an accident caused by a self-driving car or drone, there may be questions about who is liable for the damage. There is a need to establish a legal framework that can assign responsibility and liability in the event of accidents caused by self-driving cars (Joffe-Block, 2018).
- Data protection and ownership: The use of cameras and sensors in self-driving cars raises concerns about privacy and data protection (Crawford and Calo, 2016).
- Regulation: The legal issue of regulation involves the development and enforcement of laws and regulations governing self-driving cars. This includes issues such as safety standards, licensing requirements, and insurance policies.
- Intellectual Property: The legal issue of intellectual property involves the protection of patents, trademarks, and copyrights related to self-driving car technology. This includes issues of ownership, licensing, and infringement.
- Regulation: The legal issue of regulation involves the development and enforcement of laws and regulations governing self-driving cars. This includes issues such as safety standards, licensing requirements, and insurance policies.
- Intellectual Property: The legal issue of intellectual property involves the protection of patents, trademarks, and copyrights related to self-driving car technology. This includes issues of ownership, licensing, and infringement.

3.6.2 Ethical Issues

- Bias and fairness: There are risks that the object detection and tracking system may be biased towards certain groups of people or objects, leading to unfair treatment. It is essential to ensure that the system is designed and trained to be fair and unbiased (Caliskan et al., 2017).
- Transparency and explainability: The system's decision-making process may be opaque and difficult to understand, making it difficult to ensure that the system is making fair and unbiased decisions. It is essential to ensure that the system's decision-making process is transparent and can be easily explained (Jobin et al., 2019).
- Safety: The ethical issue of safety involves the potential risks and benefits of self-driving cars. The possible consequences include accidents, injuries, and fatalities.

- Social Acceptance: The ethical issue of social acceptance involves the public perception of self-driving cars and the people's trust in the technology. The potential consequences include resistance to adoption, lack of support, and loss of investment.
- Decision Making: The ethical issue of decision making involves the programming of self-driving cars to make ethical decisions in certain situations. The potential consequences include dilemmas related to the value of human life, the decision-making process, and accountability.

3.6.3 Social Issues

- Job displacement: Self-driving cars have the potential to displace human workers, particularly those in the transportation industry (Frey and Osborne, 2017). Considering the potential impact of self-driving cars on employment is essential, and it is crucial to develop policies and strategies to mitigate any negative effects.
- Digital divide: The social issue of digital divide involves the unequal access to technology and the potential impact of self-driving cars on social inequality. This includes issues of access to technology, digital literacy, and economic disparities.
- Quality of life: The social issue of quality of life involves the potential impact of self-driving cars on people's well-being and quality of life. This includes issues of health, safety, and environmental quality.
- Social interaction: The social issue of social interaction involves the potential impact of self-driving cars on human interaction and social behavior. This includes issues of communication, social norms, and cultural practices.
- Social inclusion: The social issue of social inclusion involves the potential impact of self-driving cars on social inclusion and diversity. This includes issues of equity, justice, and social cohesion.
- Community engagement: The social issue of community engagement involves the potential impact of self-driving cars on community engagement and civic participation. This includes issues of public involvement, community empowerment, and democratic governance.
- Cultural change: The social issue of cultural change involves the potential impact of self-driving cars on cultural practices, values, and norms. This includes issues of identity, tradition, and social change.

4. Results

In this section, the result of of the models are presented. The Udacity dataset has 15,000 images, with 12,000 for training and 3,000 for validation. The model was trained and the results was present in the subsections below.

4.1 Object recognition

The Precision of each model as expressed by equation (3.3) is shown in Figure 4.1(a). Precision measures the proportion of correctly identified objects out of all positive predictions. In this case, YOLOv3 has a precision of 0.69, which means that out of all the objects it predicted, 69% were correct. On the other hand, YOLO-R has a precision of 0.62, indicating that it correctly predicted 62% of the objects.

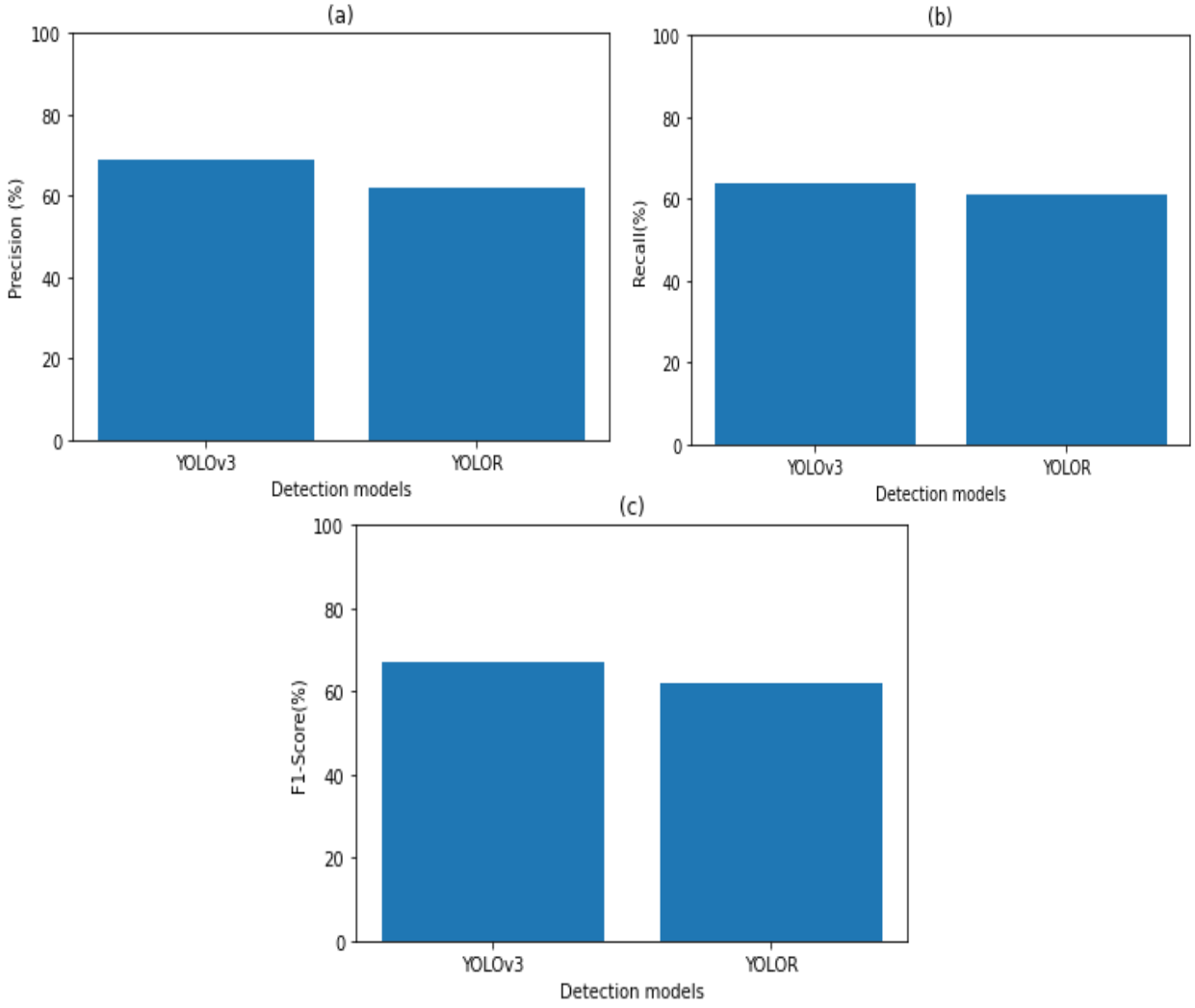

Figure 4.1: Model comparison for (a) precision (b) Recall (c) F1-score

Recall is a metric used to evaluate the performance of a recognition model, and it represents the proportion of correctly identified objects out of the total objects that should be identified by the model. It is expressed in Equation (3.2) and shown for each model in Figure 4.1 (b).

YOLOv3 has a recall of 0.64, which means that it correctly identified 64% of the actual objects in the dataset. YOLO-R, on the other hand, has a recall of 0.61, indicating that it correctly identified 61% of the objects.

The F1-score as expressed in Equation (3.4) and presented in Figure 4.1(c) is a metric that balances Precision and recall by taking their harmonic mean. It provides a single score that summarises the overall performance of the model. YOLOv3 has an F1-score of 0.67, while YOLO-R has an F1-score of 0.62. This suggests that YOLOv3 performs better overall in terms of both Precision and recall.

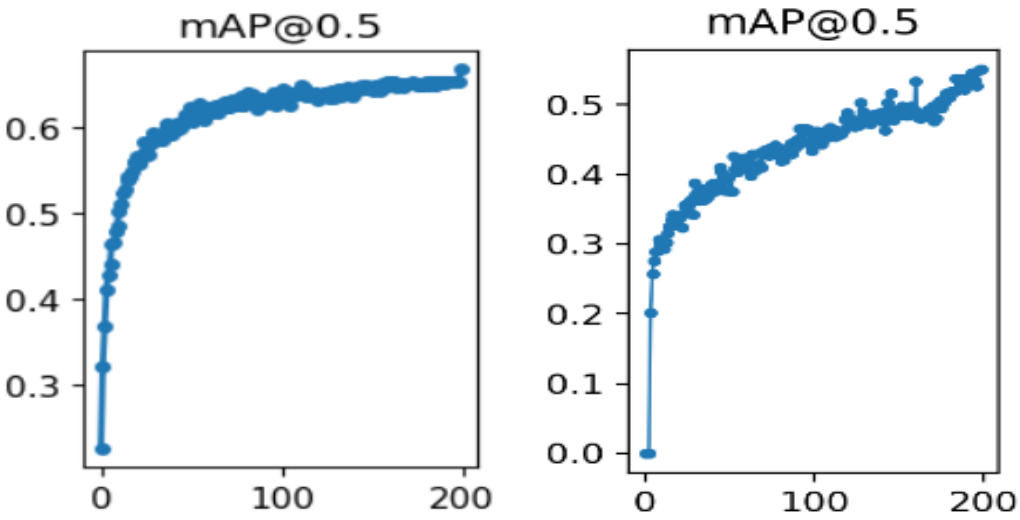

Figure 4.2: mean average Precision for YOLOV3 (right), and YOLOR (left)

To evaluate the performance of recognition models quantitatively, we measured mAP (mean average Precision) as depicted in Figure 4.2. The area under the precision-recall curve is calculated to determine the AP, and the mAP is obtained by averaging all APs in all classes. From the results presented in Figure 4.2, YOLOv3 achieved the highest identification performance, making it the most suitable model for object recognition in this scenario.

4.2 Object Tracking

To evaluate the tracking accuracy of objects, such as cars, we have defined the tracking precision as Equation (3.5). This metric represents the percentage of maintaining the same tracking ID for each object in successive frame images. Figure 4.3 illustrates the tracking precision of the models for each class. Based on the results, YOLOv3 demonstrates reasonable accuracy in recognising various object classes in different environments. To further validate the performance of the developed models, inferences were conducted on driving videos, and the results were presented in frames under different driving conditions, as depicted in Figure 4.4 to Figure 4.7.

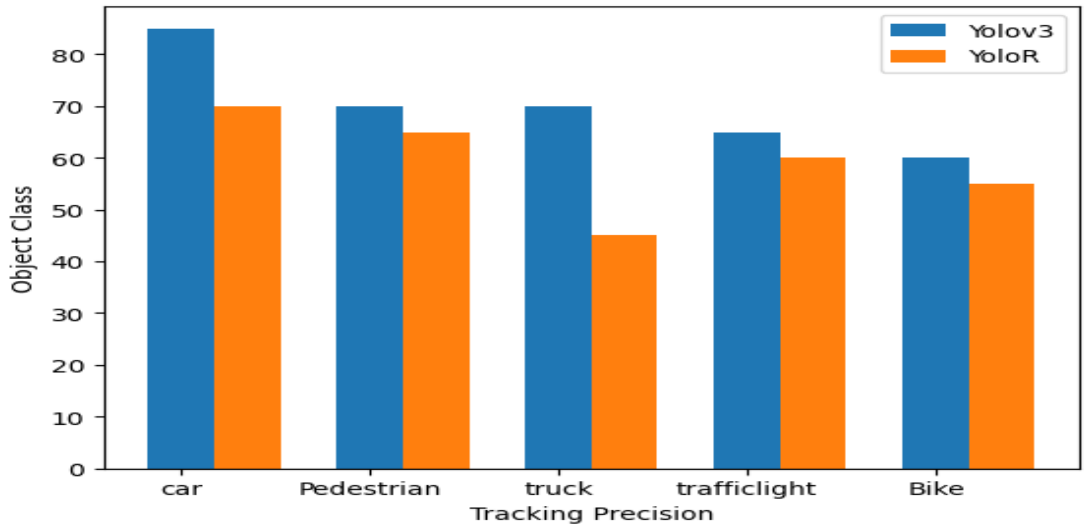

Figure 4.3: TP of models for each class.

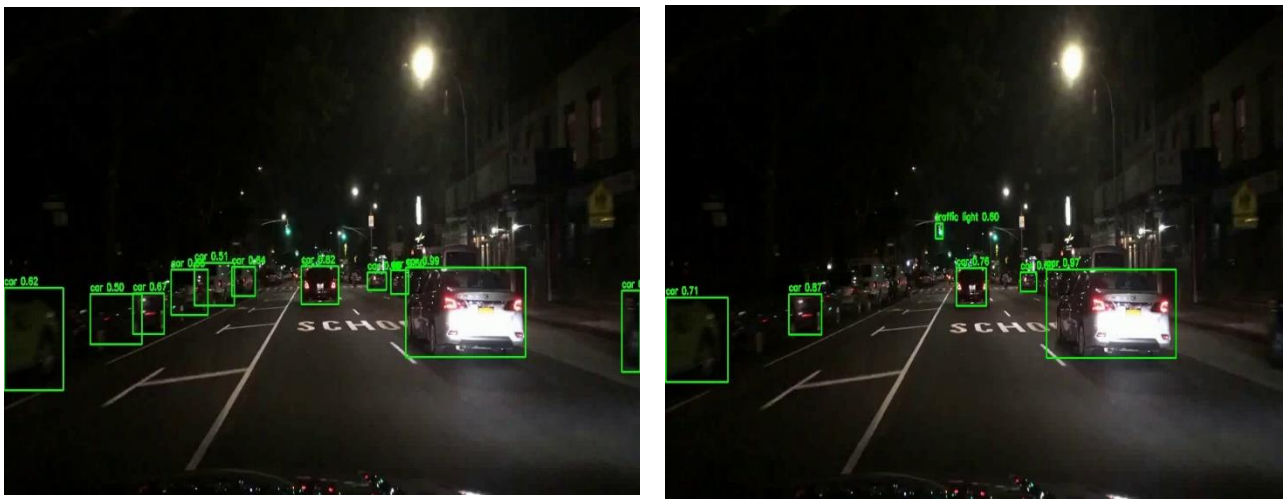

Figure 4.4: The test case for object detection for YOLOv3 (Right) and YOLO-R(Left) at frame 0.

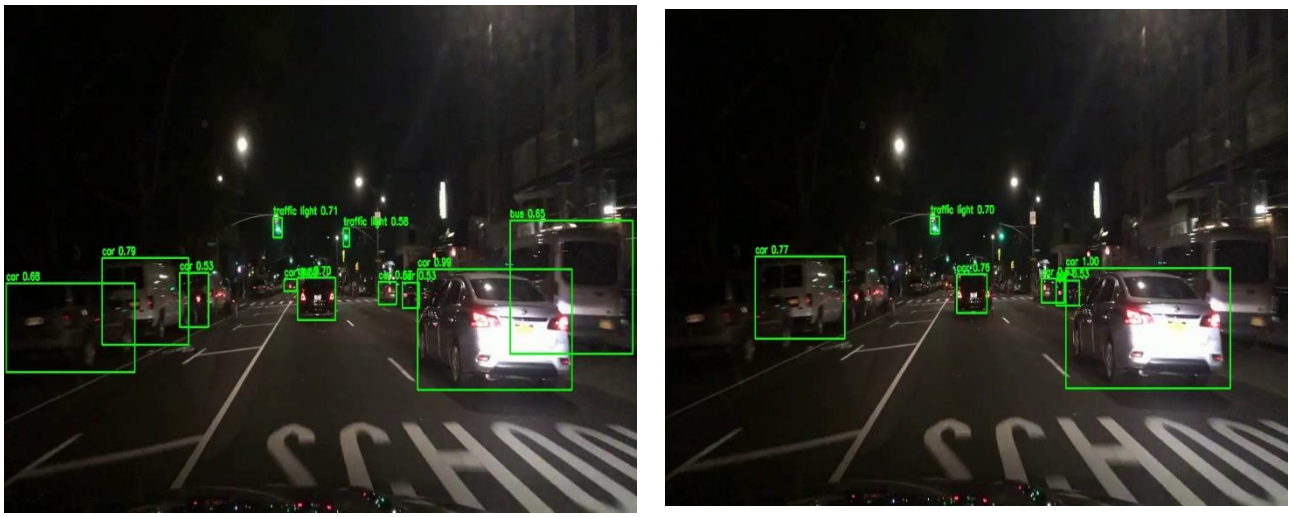

Figure 4.5: The test case for object detection for YOLOv3 (Right) and YOLO-R(Left) at frame 30.

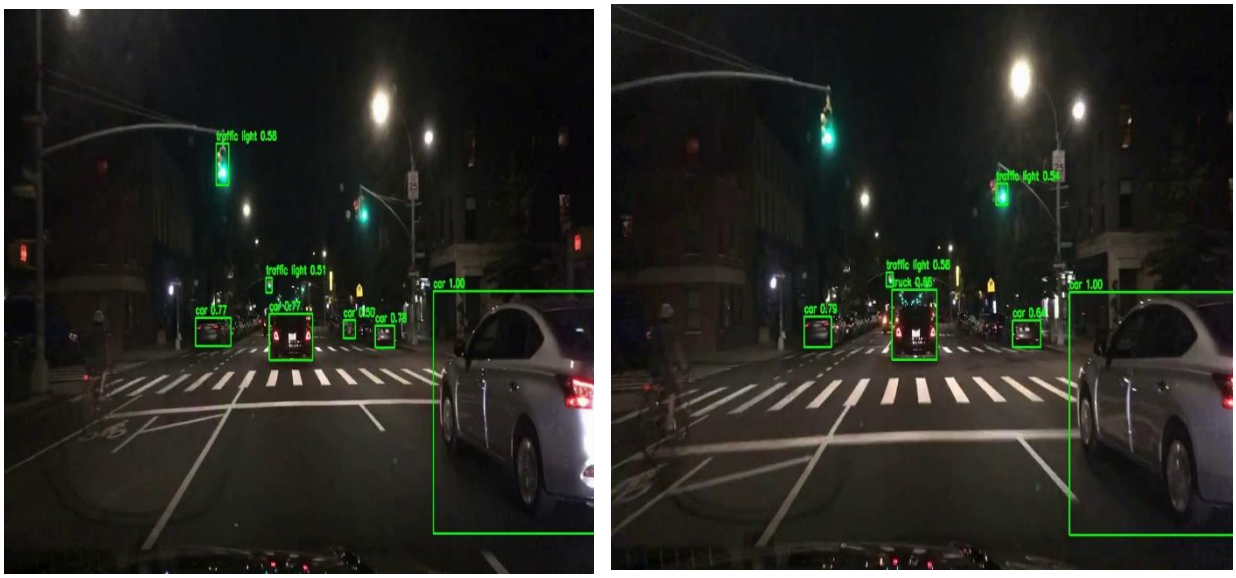

Figure 4.6: The test case for object detection for YOLOv3 (Right) and YOLO-R(Left) at frame 120

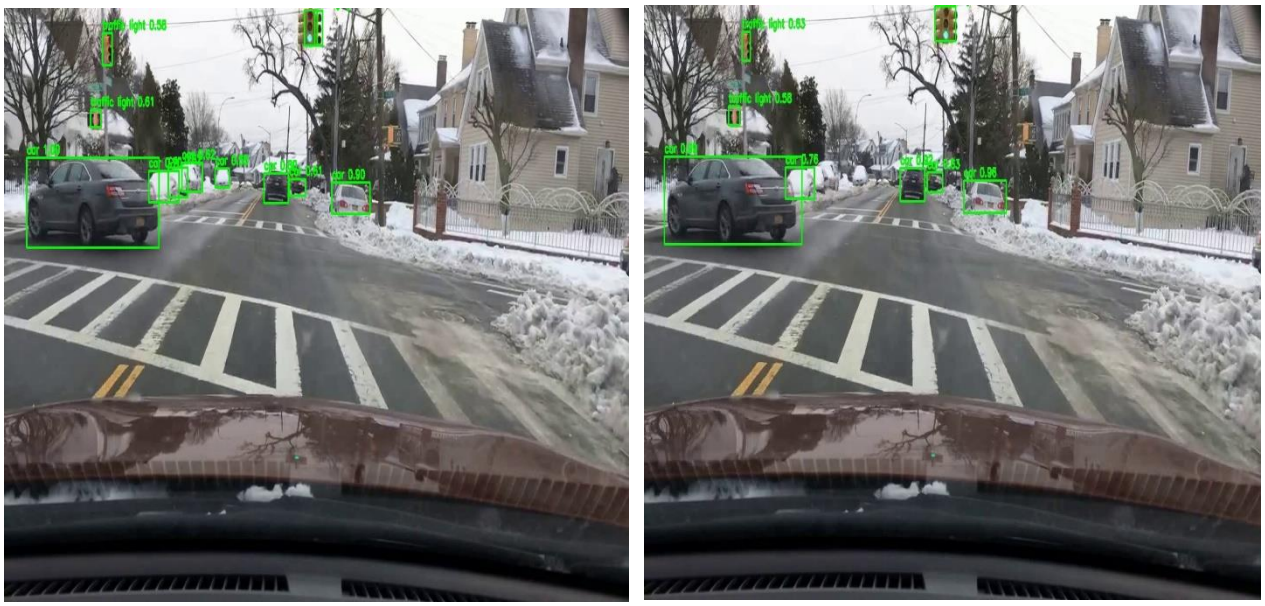

Figure 4.7: The test case for object detection for YOLOv3 (Right) and YOLO-R(Left) at frame 300.

5. Discussion

Self-driving cars have the potential to transform the way we commute and travel. One of the key challenges in developing safe and effective self-driving cars is the ability to accurately detect and track objects in the vehicle's environment. In this study, deep learning techniques was explored to develop an object detection and tracking system for self-driving cars.

To answer the first question; Can deep learning techniques improve the performance of self-driving car?

- Yes; we developed and optimised two (2) deep learning models for object detection and tracking.

To answer the second question; Can the proposed system handle small and occluded objects, changes in lighting and weather conditions, and can it be robust to these variations?

- Yes; the developed and optimised models for object detection and tracking can handle challenging driving scenarios, such as low light, heavy traffic, and adverse weather conditions. We trained and validated our models on a large dataset of annotated images and videos of various driving scenarios.

To answer the third question, How does the proposed system compare to state-of the-art object detection and tracking methods in terms of performance and real-time capabilities?

- Evaluation metrics was used to measure the performance of the proposed system as shown in Figure 4.1-Figure 4.7. It is vital to compare the develop system with state-of-the art, the system performs better than several others as represented in Figure 5.1.

5.1 Performance Evaluation

This study presented Precision, a measure of accuracy, recall, and F1-score, in Figure 4.1, showing that YOLOv3 outperformed YOLO-R in accuracy and prediction. Although these three metrics are important, a trade-off is necessary. For instance, accuracy and prediction time are essential in medical imaging and industrial quality control, while Precision is critical in advanced driver assistance systems (ADAS), which require object detection to identify and track vehicles, pedestrians, and other objects in real-time. The results indicated that YOLOv3 detected small and occluded objects with higher Precision, recall, and F1-score values compared to YOLO-R in the Udacity dataset. This finding implies that YOLOv3 is better suited for object detection tasks that require high levels of accuracy and Precision, such as autonomous vehicles. YOLOv3's higher precision and recall values may also make it more reliable and effective in these applications by detecting and tracking objects more accurately and consistently than YOLO-R. Additionally, YOLOv3 showed slightly higher Precision on the Udacity dataset than YOLO-R, although the difference was minimal. It is worth noting that tests performed on small datasets like Udacity may not represent overall performance as effectively as tests on larger datasets like COCO and Pascal VOC.

To test the ability of the proposed system to handle small and occluded objects, changes in lighting and weather conditions (Research question 2), the study presented Figure 4.4 to Figure 4.7. In one of the frames, the pedestrian and the bike were undetectable by the two models, likely due to low image resolution, scene complexity, or other factors. However, YOLOv3 was better at detecting occluded objects and performed better than YOLO-R.

5.2 Mean Average Precision (mAP) Comparison

In the field of object detection, the YOLOv3 model has become a popular choice due to its high accuracy and computational efficiency. Two recent studies by Zhao-Zhao et al. (2020) and Naftali et al. (2022) explored different approaches to optimise the YOLOv3 model for object detection tasks. Zhao-Zhao et al. (2020) employed the K-means clustering algorithm to analyse the anchor number and aspect ratio of the Udacity dataset, resulting in a more suitable parameterisation of their model. They also added a 104 x 104 feature detection layer to improve the performance of the road target detection algorithm, achieving an impressive mAp of 78.83.

Naftali et al. (2022) took a different approach and modified the Udacity dataset by rescaling, hue shifting, and adding noise to the images. Their YOLOv3 model achieved a respectable mAp of 0.583 on this modified dataset.

Meanwhile, the original YOLOv3 model by Redmond and Farhadi (2018) is a versatile object detection model that can detect a wide range of objects with high accuracy. YOLOv3-320 achieved a mAp of 51.5, YOLOv3-416 achieved a mAp of 55.3, and YOLOv3-608 achieved a mAp of 57.9. These variants with different input image sizes can impact the model's accuracy and computational requirements.

In this study, we fine-tuned a pre-trained YOLOv3 model on custom datasets for object detection tasks, with a mAp of 0.66, capable of detecting a wide range of objects. These findings demonstrate the versatility and adaptability of the YOLOv3 model and provide insights into different approaches to optimise its performance for object detection tasks. The results from this study are among the top classifier as seen in Figure 5.1. This performance can be attributed to Transfer learning which involves using a pre-trained model as a starting point for training a new model on a different task or dataset.

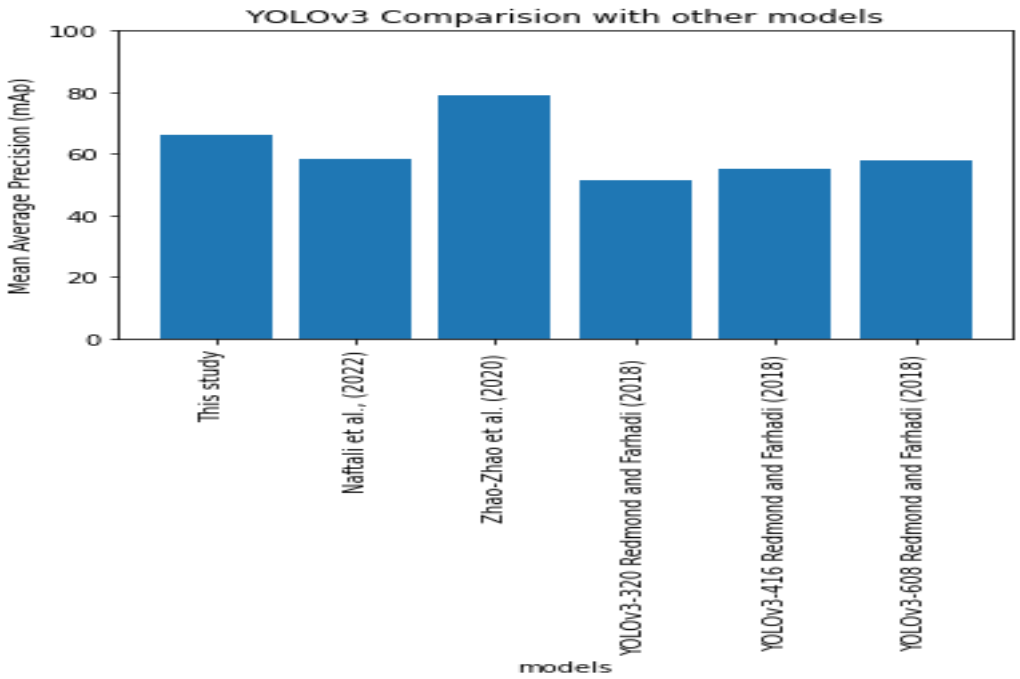

Figure 5.1: mAP comparison of the tuned algorithm with the state-of-the-art.

6.0 Conclusion

In this study, an effective method in object detection and classification dataset has been proposed. To advance the forefront object detection, we conducted experiments using the Udacity dataset and found significant performance improvements compared to previous studies. Based on the results, it can be concluded that YOLOv3 outperforms YOLO-R on the Udacity dataset in terms of Precision, recall, and F1-score. This suggests that YOLOv3 may be a more reliable and effective choice for object detection tasks requiring high accuracies and Precision, such as autonomous vehicles. The top-performing algorithms (YOLOv3) show higher mean average Precision than the forefront. However, it is essential to note that the Udacity dataset is limited in scope and may not fully capture the complexities and variability of real-world object detection scenarios. Therefore, the generalizability of these results to other domains and datasets is uncertain, and caution should be taken when applying these findings to different contexts.

6.1 Critical Evaluations and Future works

6.1.1 Achievement and Self Reflection

- **Achievements**

The project set out to develop and optimise deep learning models for object detection and tracking that can handle challenging driving scenarios, such as low light, heavy traffic, and adverse weather conditions. To achieve this, the study evaluated the performance of the developed system through extensive testing and validation using performance and evaluation metrics. Additionally, the study compared the developed system against existing forefront object detection and tracking systems for self-driving cars and identified areas for further improvement.

The study proposed an effective method for object detection and classification using the Udacity dataset, which showed improved performance compared to previous studies. The study found that YOLOv3 outperforms YOLO-R in terms of Precision, Recall, and F1-score, suggesting that it may be a more reliable and effective choice for object detection tasks requiring high accuracy levels and Precision.

Overall, the project achieved the objectives set out at the beginning, providing valuable insights into developing and optimising deep learning models for detecting and tracking object in challenging driving scenarios. The study successfully developed and optimised deep learning models that were capable of handling challenging driving scenarios, such as low light, heavy traffic, and adverse weather conditions. The performance of the developed system was extensively tested and validated using performance and evaluation metrics. The study's findings will be useful for further research in developing object detection and tracking systems for self-driving cars.

- **Self-Reflection**

Reflecting on my experience conducting this research, I must acknowledge that I faced several difficulties and challenges along the way. One of the most significant challenges was selecting the appropriate algorithm, training platform, and optimising its hyperparameters. To address this challenge, I read several research papers to establish the best algorithm for the project. I also utilised trial and error to fine-tune the hyperparameters and improve the model's performance.

Another challenge I encountered was the limited scope of the Udacity dataset, which presented potential limitations in the generalizability of the results. I employed various techniques to mitigate this issue, such as data augmentation and transfer learning, to enhance the model's performance and address the potential for overfitting.

Overall, I believe that this research was a rewarding experience that allowed me to learn and grow. While I encountered several challenges, I was able to overcome them through perseverance, collaboration, and continuous learning. I would like to explore other deep learning algorithms, ensemble methods, and transfer learning techniques to improve object detection and classification performance.

6.1.2 Recommendations

The study's findings and objectives provide a basis for several research recommendations in the field of object detection and tracking for self-driving cars. First, it is recommended to extend the evaluation of the proposed method to other datasets and domains to investigate its generalizability and identify potential limitations. Moreover, the proposed method's ability to handle complex and dynamic scenarios, such as crowded and cluttered environments, occlusion, and fast-moving objects, can be evaluated. Finally, conducting a user study to evaluate the usability and effectiveness of the proposed method in real-world applications, such as autonomous vehicles, medical imaging, and industrial quality control, is a promising research direction. These recommendations can contribute to developing more reliable and effective autonomous systems and advance the research in object detection and tracking for self-driving cars.

6.1.3 Future Works

Based on the study's objectives and findings, several future works can be pursued. First, further improving the proposed system's performance and efficiency can be explored by optimising its architecture and hyperparameters. The proposed system's scalability and compatibility with different hardware and software platforms should also be evaluated to facilitate its deployment in various autonomous vehicle systems. Furthermore, conducting real-world experiments to evaluate the system's performance and safety in different driving scenarios and environments can provide insights into its practical feasibility and limitations. These future works can contribute to the development of more advanced and practical autonomous vehicle systems that can improve road safety and transportation efficiency.

REFERENCES

Agafonov, A. and Yumaganov, A., 2020, May. 3D Objects Detection in an Autonomous Car Driving Problem. In *2020 International Conference on Information Technology and Nanotechnology (ITNT)* (pp. 1-5). IEEE.

Al-refai, G. and Al-refai, M., 2020. Road Object Detection using Yolov3 and Kitti Dataset. *International Journal of Advanced Computer Science and Applications*, *11*(8).

Amanatiadis, A., Karakasis, E., Bampis, L., Ploumpis, S. and Gasteratos, A., 2019. ViPED: On-road vehicle passenger detection for autonomous vehicles. *Robotics and Autonomous Systems*, *112*, pp.282-290.

Arnold, E., Al-Jarrah, O.Y., Dianati, M., Fallah, S., Oxtoby, D. and Mouzakitis, A., 2019. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), pp.3782-3795.

Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B., 2016, September. Simple online and real-time tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). IEEE.

Bourennane, M., 2022. *Visual Object Tracking Approach Based on Wavelet Transforms* (Doctoral dissertation, Faculté des Sciences et de la technologie).

Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Boukerche, A. and Hou, Z., 2021. Object detection using deep learning methods in traffic scenarios. *ACM Computing Surveys (CSUR)*, *54*(2), pp.1-35.

Castelló, V.O., Igual, I.S., del Tejo Catalá, O. and Perez-Cortes, J.C., 2020. High-profile VRU detection on resource-constrained hardware using YOLOv3/v4 on BDD100K. *Journal of imaging*, *6*(12).

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), pp.834-848.

Chen, Z. and Huang, X., 2019. Pedestrian detection for autonomous vehicle using multi-spectral cameras. *IEEE Transactions on Intelligent Vehicles*, *4*(2), pp.211-219.

Chen, J. and Bai, T., 2020. SAANet: Spatial adaptive alignment network for object detection in automatic driving. *Image and Vision Computing*, *94*, p.103873.

Deng, Y., Zhang, T., Lou, G., Zheng, X., Jin, J. and Han, Q.L., 2021. Deep learning-based autonomous driving systems: A survey of attacks and defenses. *IEEE Transactions on Industrial Informatics*, *17*(12), pp.7897-7912.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L. and Zou, H., 2018. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, *145*, pp.3-22.

F. Chollet et al. Keras. https://keras.io, 2015.

Fujiyoshi, H., Hirakawa, T. and Yamashita, T., 2019. Deep learning-based image recognition for autonomous driving. *IATSS research*, *43*(4), pp.244-252.

Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? Technological Forecasting and Social Change, 114, 254-280.

Grigorescu, S., Trasnea, B., Cocias, T. and Macesanu, G., 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, *37*(3), pp.362-386.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

Grewal, M.S., Andrews, A.P. and Bartone, C.G., 2020. Kalman filtering.

Han, S., Mao, H. and Dally, W.J., 2015. Deep compression: Compressing deep neural networks with pruning, trained quantisation and huffman coding. *arXiv preprint arXiv:1510.00149*.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hou, X., Wang, Y. and Chau, L.P., 2019, September. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

Kosaka, N. and Ohashi, G., 2015. Vision-based nighttime vehicle detection using CenSurE and SVM. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), pp.2599-2608.

Khan, A.M., 2021. *Vehicle and pedestrian detection using YOLOv3 and YOLOv4 for self-driving cars* (Doctoral dissertation, California State University San Marcos).

K. F. Chanho, L. A. Ciptadi, and J. Rehg. 2018, "Multiple Hypothesis Tracking Revisited". In: (2018). doi: 10.1109/WACV.2018.00087.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.

Liu, G., 2017. *Real-time Object Detection for Autonomous Driving Based on Deep Learning* (Doctoral dissertation, Texas A&M University-Corpus Christi)

Lidestam, B., Thorslund, B., Selander, H., Näsman, D. and Dahlman, J., 2020. In-Car Warnings of Emergency Vehicles Approaching: Effects on Car Drivers' Propensity to Give Way. *Frontiers in Sustainable Cities*, *2*, p.19.

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

Mobahi, M. and Sadati, S.H., 2020, August. An Improved Deep Learning Solution for Object Detection in Self-Driving Cars. In *2020 28th Iranian Conference on Electrical Engineering (ICEE)* (pp. 1-5). IEEE.

Mai, N.A.M., Duthon, P., Khoudour, L., Crouzil, A. and Velastin, S.A., 2021. 3D object detection with SLS-fusion network in foggy weather conditions. *Sensors*, *21*(20), p.6711.

Mirza, M.J., Buerkle, C., Jarquin, J., Opitz, M., Oboril, F., Scholl, K.U. and Bischof, H., 2021, September. Robustness of object detectors in degrading weather conditions. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 2719-2724). IEEE.

Masmoudi, M., Ghazzai, H., Frikha, M. and Massoud, Y., 2019, September. Object detection learning techniques for autonomous vehicle applications. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (pp. 1-5). IEEE.

Naftali, M.G., Sulistyawan, J.S. and Julian, K., 2022. Comparison of Object Detection Algorithms for Street-level Objects. *arXiv preprint arXiv:2208.11315*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A., 2017. Automatic differentiation in pytorch.

Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tao, Q., Zhang, L., & Wang, S., 2021). Real-time multi-object tracking in self-driving cars. In 2021 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 1-6).

Wang, X., Zhao, Y. and Pourpanah, F., 2020. Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, *11*, pp.747-750.

Wei, L., Li, Z., Gong, J., Gong, C. and Li, J., 2021, September. Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 44-51). IEEE.

Welch, G.F., 2020. Kalman filter. *Computer Vision: A Reference Guide*, pp.1-3.

Wojke, N., Bewley, A. and Paulus, D., 2017, September. Simple online and real-time tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE.

Yudin, D.A., Skrynnik, A., Krishtopik, A., Belkin, I. and Panov, A.I., 2019. Object detection with deep neural networks for reinforcement learning in the task of autonomous vehicles path planning at the intersection. *Optical Memory and Neural Networks*, *28*, pp.283-295.

Zhao-zhao, J.I.N. and Yu-fu, Z.H.E.N.G., 2020, October. Research on application of improved YOLO V3 algorithm in road target detection. In *Journal of Physics: Conference Series* (Vol. 1654, No. 1, p. 012060). IOP Publishing.

Zhang, H., Xu, T., Li, H., Zhang, S., & Huang, X., 2021. Deep Learning-Based Multiple Object Tracking: A Review. IEEE Transactions on Intelligent Transportation Systems, 1-16.

Zhou, X., Wang, D. and Krähenbühl, P., 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.